

Desenvolupament de l'aplicació web per una experiència gastronòmica

David Latorre

Resum— El propòsit general del projecte és el de desenvolupar una pàgina web per recolzar un projecte de posada en marxa d'un restaurant, que actuara com a client, adaptant el desenvolupament del projecte a les necessitats del client al llarg del projecte. Tambè es presentaran i explicaran els diferents estudis, opcions, propostes, aspectes, etc. que s'han tingut en compte i com han afectat a la hora de dur a terme el desenvolupament del mateix.

Paraules clau— Angular, TypeScript, Firebase, CSS, HTML, front-end, aplicació web.

Abstract— The general purpose of the project is to develop the web page for a gastronomic experience, which will act as the client side of the project, adapting the development of it to the client's needs throughout the whole duration. Also, all different studies, options, proposals, aspects, etc. That have been part of or affected the project development will be presented and discussed.

Keywords— Angular, TypeScript, Firebase, CSS, HTML, front-end, web page.

1 INTRODUCCIÓ - CONTEXT DEL TREBALL

EN aquest document i d'ara endavant, es parlarà de la figura de client del projecte, que en aquest cas, és el propietari o impulsador del projecte "Il gioco dell'amore", un lloc de reunió i d'oci entorn d'una experiència gastronòmica d'estil italià, de pizzeria tradicional napolitana més concretament.

En el moment d'inici d'aquest projecte, el "projecte client" es troba en fase de començament o *beta*, en el que actualment l'experiència està reservada a familiars, amics i coneguts propers, però l'objectiu és començar a obrir-se al públic gradualment, mitjançant per una banda petites reunions en grup amb algun sistema d'invitacions i reserves i per l'altre començar a oferir un servei de comandes per emportar.

El context d'aquest treball s'ha vist afectat en bona part per la situació general del moment, on actes socials d'aquest tipus s'han hagut de reduir pràcticament en la seva totalitat i ha fet que les necessitats o objectius tant del client com

d'aquest projecte variïn en conseqüència per adaptar-se als canvis que la situació requeria. A continuació en aquest document es detallen els objectius, l'estat de l'art, la metodologia aplicada, la planificació, el desenvolupament, resultats, línies de continuació, conclusions del projecte i bibliografia d'interès per al lector, a part de la secció d'apèndix amb figures addicionals.

2 OBJECTIUS

Els objectius d'aquest projecte van fortament lligats amb els objectius del client a l'hora d'implementar aquesta nova eina, és per això que primer necessitem conèixer el context del projecte del client i els seus objectius amb més detall.

2.1 *Il Gioco dell'Amore*

Aquest apartat vol ajudar al lector a comprendre com funciona aquesta experiència gastronòmica de la qual es parla al llarg d'aquest document.

El principal objectiu de *Il Gioco dell'Amore* és el de transportar una petita part de la gastronomia Napolitana al territori Català, en concret a Sant Quirze del Vallès, on resideix l'impulsor del projecte, de nom *Eric Ayala*, qui va quedar meravellat per l'art de la pizza i tota la cultura i tradicions que l'envolten quan va poder gaudir-ho en primera persona a Nàpols en un viatge turístic.

• E-mail de contacte: davidlatorrelazaro@gmail.com
 • Menció realitzada: Tecnologies de la Informació
 • Treball tutoritzat per: Ramón Musach Pi Departament d'Enginyeria de la Informació i de les Comunicacions, Àrea de Ciències de la Computació i Intel·ligència Artificial
 • Curs 2020/21

Des d'aquell mateix moment, el seu objectiu s'ha convertit en traslladar una part d'aquesta experiència al nostre territori, mantenint en tot moment l'essència i tradicions originals. Durant els dos últims anys, ha anat endinsant-se en la professió de *Pizzaiolo*, primer instal·lant un forn de llenya original napolità en el jardí de casa, per tal de poder practicar amb regularitat, després realitzant un curs oficial organitzat per la *Associazione Verace Pizza Napoletana*, que l'acredita com a autèntic pizzaiolo napolità. A partir d'aquest moment, es comença a desenvolupar l'experiència gastronòmica en la qual es basa el projecte.

Aquesta consisteix en oferir un servei de restaurant, a petita escala, on cada dissabte s'obre un servei de "Take Away" durant tota la tarda i el vespre. També hi ha una intenció a mitjà termini d'oferir el servei en el mateix local, amb l'aforament molt limitat i respectant les mesures de seguretat imposades. Per tal d'oferir una experiència tradicional de qualitat, tots els ingredients utilitzats provenen directament d'Itàlia, principalment de la regió de Campania d'on es origina aquesta tradició, i són els mateixos utilitzats per les pizzeries tradicionals Napolitanes més importants.

La base d'usuaris inicial són un grup reduït de coneguts, amics i familiars, que s'espera anar ampliant gradualment a mesura que la gent gaudeixi de l'experiència i ho recomani en el seu entorn. Tot això es fa amb l'objectiu a llarg termini d'obrir un restaurant en plenes condicions en uns anys, quan s'hagi adquirit molt més grau d'experiència i la situació sanitària que afecta aquest tipus d'activitats hagi millorat respecte a l'actual.

2.2 Captació de Requisits

Aquesta subsecció explica els detalls de la **tasca 14**, que apareix en la planificació inicial de la figura 9, explicada més endavant, i que ajuda a consolidar els objectius i requisits del projecte.

En aquesta fase es vol establir els requisits de l'aplicatiu. Es fa una entrevista amb el client, on es pregunten les seves intencions, i s'intenta copsar què és el que realment necessita d'aquest projecte. A partir d'aquí es determinen uns requisits, funcionals i no funcionals, sobre els que s'aniran desenvolupant diferents parts de la web. A més a més, també es decideix un ordre de prioritats d'aquests requisits, segons criteri del client, per tal de desenvolupar-los o implementar-los de forma gradual si fos necessari i poder rebre el *feedback* adequat en reunions de seguiment posteriors. La llista de requisits inicial de l'aplicació, ordenats segons prioritat i explicats breument, queda així:

- Pàgina d'Informació - Oferir la informació de l'experiència a qualsevol persona que vegi la web, que serveixi de presentació i faciliti el seu coneixement. Ha de mostrar la informació que el client desitgi i intentar fer-ho de la millor forma possible quant a l'aspecte visual.
- Pàgina de contacte - Oferir una pàgina dedicada a que potencials usuaris es posin en contacte amb el client, per demanar informació extra, solucionar dubtes, requerir algun servei, etc.
- Disponibilitat en tots els dispositius - Es demana que l'eina funcioni en qualsevol classe de dispositiu, sense importar el seu sistema operatiu o el model, per tal que qualsevol usuari potencial pugui utilitzar-la.
- Desenvolupar un sistema de comandes en línia - És un dels requisits principals de la web, per tal de gestionar de forma automatitzada aquestes comandes i recolzar el sistema actual a través de trucades telefòniques.

- Desenvolupar un sistema de reserves en línia - És un altre dels requisits principals, orientat a gestionar les reserves en el local físic de forma automatitzada.
- Generar informes o llistes - Per tal de portar un millor control en el desenvolupament de les activitats, desenvolupar un sistema que permeti al client obtenir informació d'aquest estil de forma ràpida i concentrada. Per exemple una pàgina on el client pugui veure les comandes actuals en espera.
- Desenvolupar un sistema basat en invitacions - Orientat a obrir a poc a poc l'abast del projecte, però mantenint una certa exclusivitat. Aquest sistema ha de ser flexible a canvis

A més a més, a causa de la naturalesa del treball, hi ha una sèrie d'objectius inherents com són els de realitzar un treball en condicions similars al món professional real, adquirir una base de coneixements i desimboltura en el desenvolupament web, formar una experiència en la gestió d'un projecte, la seva planificació i el seu cicle de vida en general.

3 ESTAT DE L'ART

Degut a alguns dels objectius i necessitats que s'han presentat en anteriors apartats, la investigació en l'estat actual de l'art s'ha centrat en les eines multiplataforma. Avui en dia es pot observar una gran tendència a desenvolupar aplicacions amb aquest tipus d'eines, és per això que hi ha moltes opcions per dur a terme el desenvolupament d'aquestes tecnologies i per tant haurem de considerar els diferents avantatges que ens aportin. A continuació es detalla les principals opcions que s'han considerat:

- Desenvolupar una aplicació mòbil nativa: una opció seria desenvolupar una aplicació per android/iOs ja sigui amb els seus respectius llenguatges o amb llenguatges multiplataforma innovadors, com per exemple kotlin multiplatform mobile, enfocat a desenvolupar aplicacions multiplataforma, principalment iOS i Android. A més a més caldria implementar un esquema client-servidor al backend
- Desenvolupar una pàgina web: la segona opció seria desenvolupar una pàgina web, on l'accés per part de l'usuari i potencial client és més ràpid i senzill, sense descàrregues pel mig.
- Desenvolupar una "progressive web app" (PWA) [1]: aquesta opció és bàsicament una millora de la segona opció, on es desenvoluparia una pàgina web multiplataforma i més tard s'implementaria els serveis que fan una PWA una de les principals vies de desenvolupament en les empreses d'avui en dia.

Després d'estudiar les tres opcions, considerem que la segona és la millor opció, ja que satisfà en millor mesura els interessos del client i a més a més pot millorar-se cap a la tercera opció en un futur, un cop la base de l'aplicació ja estigui consolidada

Un cop presa aquesta decisió, hem d'investigar el gran conjunt d'opcions que disposem a l'hora de desenvolupar una web d'aquestes característiques. Investigant una mica sobre les tendències actuals en el desenvolupament de pàgines web, observem principalment dues opcions.

- Frontend frameworks amb Node.js/JavaScript Backends: Desenvolupar la pàgina web mitjançant un framework com React, Angular o Vue, que treballen amb llenguatges html, typescript, css, etc per desenvolupar el frontend, juntament amb el backend en Node.js, allotjat en algun servei de *hosting* com Hostinger, Wix o Firebase
- CMS: La segona opció seria la d'integrar tota l'eina sota el sistema de gestor de continguts Wordpress [4], que ofereix tots els serveis necessaris per desenvolupar una pàgina web, com seria la gestió de dominis, allotjament i disseny de la

web. Actualment és un dels recursos més utilitzats a l'hora de desenvolupar pàgines web, ja que ofereix gran quantitat de serveis per facilitar la tasca, segons W3Tech [5] aproximadament un 41'5% de les pàgines web actuals utilitzen WordPress.

D'aquestes dues opcions, s'ha triat la primera, apostant per desenvolupar la pàgina de forma completa, utilitzant Angular [2], un framework de programació en Typescript, de codi obert i mantingut per Google, per el que ofereix una àmplia integració amb els seus serveis. A més a més s'utilitzarà Firebase [3], una plataforma de Google pensada desenvolupar aplicacions mòbils i web de forma més ràpida i senzilla, ja que ofereix serveis com *hosting*, base de dades en temps real, gestió d'autoritzacions, integració amb Google Analítics i molts altres serveis de forma gratuïta, sempre que no es superin els mínims que estableix.

Es considera que mitjançant aquestes dues eines, es pot desenvolupar una aplicació web potent i ben integrada i proporcionen eines més que suficients amb la avantatge de ser gratuïtes respecte al cost mensual de WordPress.

4 METODOLOGIA

El projecte s'ha dut endavant aplicant-hi una Metodologia de tipus *Agile* [6], basada en diversos principis bàsics, com per exemple la planificació incremental, és a dir que es poden anar afegint o canviant requisits per adaptar el projecte i això més que un problema suposa una avantatge competitiva. També s'ha de mantenir en tot moment el client satisfet i informat sobre els aspectes rellevants del projecte i el desenvolupament es du a terme en petits cicles, on cada dues setmanes es presentaran les funcionalitats ja desenvolupades al client, per tal de rebre suggeriments de canvis i/o millora com més aviat millor, per planificar l'execució a curt i mitjà termini (de cara a les següents entregues i reunions) i assegurar que el projecte va pel camí adient.

4.1 Planificació Inicial

En el moment inicial del projecte es realitza una primera planificació amb la intenció de reflectir els passos a seguir per a desenvolupar el projecte, però aquesta no és estricta, ja que com s'ha comentat abans, la planificació ha anat variant a mesura que avançava el desenvolupament del projecte, per adaptar-se als diferents canvis que hi ha hagut en les necessitats del client o en els objectius de l'aplicatiu i que s'expliquen més endavant en aquest document.

Aquesta primera planificació assenta les bases sobre les quals es dura a terme el desenvolupament del projecte, sobretot en quant a l'ordre de prioritats, des de les parts que es consideren més urgents i subjectes a possibles canvis fins a les parts més prescindibles o relacionades amb el tancament del projecte.

En el diagrama de Gantt de la figura 9 podem observar la línia de temps esperada del projecte. Els temps que consten en aquesta primera planificació són una estimació basada en suposicions en el moment inicial del projecte i s'ha intentat incloure una mica de marge per tal de reduir l'impacte de possibles retards en alguna fase.

5 DESENVOLUPAMENT DEL PROJECTE

En aquesta secció, s'explicarà tot el procés de desenvolupament del projecte, a nivell de codi i documentació, parts del projecte, fases, etc.

Per començar, dividirem els aspectes de desenvolupament en quatre seccions temàtiques:

- **Inici del projecte** - Aquelles tasques que es realitzen en els primers passos del projecte, prèviament al desenvolupament, per tal d'encaminar-lo correctament.

- **Firebase** - Aquelles tasques que estan directament relacionades amb els serveis de Google Firebase, que integrem en el projecte per suportar el back-end de l'aplicatiu.
- **Angular** - Les tasques relacionades amb el framework d'Angular by Google, que s'utilitza per desenvolupar tot el codi de l'aplicatiu.
- **Evolució del projecte** - Es mencionaran aquelles tasques i aspectes que s'han anat variant i replantejant al llarg del desenvolupament del projecte juntament amb les causes i conseqüències d'aquests canvis.

5.1 Inici del projecte

Les tasques realitzades en aquesta fase són les que podem veure en la figura 24, centrades en la recerca, planificació i disseny del projecte.

5.1.1 Pressupost

En aquest projecte, no s'ha realitzat un càlcul estimat del pressupost total. Això es deu al fet que es considera que el cost d'implementació el projecte és nul, com s'ha comentat en la secció d'estat de l'art es creu que hi ha suficients eines gratuïtes i de codi lliure al nostre abast en la xarxa prou potents o útils per a aconseguir dur a terme el desenvolupament sense costos addicionals.

Això és possible en gran part degut a la integració dels serveis de Google en el projecte, que sovint ofereixen plans d'ús gratuït per a desenvolupaments a petita-mitjana escala. En aquest cas, es considera que aquests plans són més que suficients per a les necessitats que hem de cobrir i en cas que fos necessari poden millorar-se (per obtenir, per exemple, funcionalitats més complexes o més amplada de banda en quant al tràfic que suporten) en qualsevol moment, ja sigui en aquest projecte o en futures línies de continuació.

5.1.2 Estudi de viabilitat

5. Estudi de viabilitat Es realitza aquest estudi per raonar i averiguar si el projecte és realitzable mitjançant diferents perspectives o factors:

- Des de la perspectiva de viabilitat de mercat, es considera que el projecte és altament viable, ja que és innegable l'augment de la tendència dels consumidors a utilitzar eines d'aquest tipus, a més a més s'hi suma la situació sanitària actual, que ha disparat els serveis de comandes en línia en el sector gastronòmic. Per altra banda, el sol fet de tenir presència en la xarxa és rellevant per al client, ja que pot ajudar a millorar la imatge del lloc i facilitar el contacte amb potencials clients.
- Des del punt de vista econòmic, s'analitzen les principals opcions plantejades després de realitzar la recerca de l'estat de l'art. S'ha arribat a la conclusió de que el projecte és viable econòmicament, ja que qualsevol de les diferents opcions plantejades són assequibles per al client, perquè la inversió és nul·la pel que fa a materials o eines necessàries per dur a terme el projecte, excepte en el cas d'haver decidit utilitzar un gestor de continguts com Wordpress per suportar el projecte, però encara que aquest fos el cas no suposaria una inversió desmesurada, sinó més aviat el contrari, ja que les tarifes oscil·len entre els 48 i 540 €/any.
- Des del punt de vista tècnic, i tenint en compte que es parteixen d'uns coneixements bàsics en el tema, es considera que l'exigència o dificultat és assequible i que desenvolupar els objectius proposats en el temps fixat és viable. A més a més es considera que es pot desenvolupar l'aplicació de forma incremental, produint primer les funcionalitats més

bàsiques i evolucionant envers les que requereixin més dedicació tècnica, sense deixar de tenir en compte les prioritats i objectius del client.

- Des del punt de vista del risc assumit en el projecte, es considera viable perquè la relació entre el risc i els possibles beneficis és prometedora. S'assumeixen pocs riscos en aquest projecte, ja que no forma part de la principal activitat del client, sinó més aviat es tracta d'un mètode per impulsar-la. En altres paraules, una fallada en el projecte no seria fatal per al client i la seva activitat, però en canvi aconseguir desenvolupar el projecte suposaria una clara millora respecte al moment actual.

5.2 Firebase

Com s'explica en la secció de l'estat de l'art, s'ha escollit integrar els serveis de back-end necessaris per al projecte amb Firebase, perquè després de realitzar una recerca en profunditat de les tecnologies de desenvolupament web disponibles s'ha considerat que és la més adequada. Firebase és una plataforma per al desenvolupament d'aplicacions web i mòbil, orientada a facilitar la creació, producció i sincronització i de nous projectes d'aquest tipus, integrat totalment en la infraestructura de Google, ofereix un gran potencial en quant a l'escalabilitat, de forma automatitzada. Ofereix tres tipus de serveis per a les aplicacions:

- Orientats a producció - Eines per facilitar el desenvolupament. Aquestes són les que principalment s'utilitzen en aquest projecte.
- Orientats al llançament i supervisió - Eines per a la gestió de tests, errors, monitoratge i mantenir l'estabilitat.
- Orientats a la participació - Eines per dur a terme una bona optimització i mantenir la satisfacció dels usuaris i adaptar-se a les seves necessitats.

En el cas d'aquest projecte, interessa especialment perquè permet aixecar el projecte sense la necessitat de gestionar un servidor propi, ja que ofereix serveis de *hosting* i gestió de domini de forma gratuïta. També cal destacar que ofereix una consola en línia amb una molt bona interfície gràfica, que facilita la gestió d'aquests aspectes del projecte de forma remota des de qualsevol dispositiu i lloc per al desenvolupador. A continuació es detallen les funcionalitats de Firebase que utilitza l'aplicació:

5.2.1 Firebase Hosting

L'aplicació està allotjada en aquest servei de *hosting* de forma gratuïta fins a un volum de tràfic limitat (360 MB/dia). El contingut se serveix en dos dominis de forma predeterminada, amb l'opció d'afegir dominis propis o personalitzats. Porta un registre de versions de llançament de l'aplicatiu de forma automàtica i ofereix la possibilitat de fer "Rollbacks" per recuperar antigues versions de forma senzilla si en algun moment és necessari. Ofereix la possibilitat de muntar canals temporals dedicats a previsualitzar els canvis i facilitar el testatge i desenvolupament. També permet veure en tot moment el tràfic de la web, l'emmagatzematge i les descàrregues totals dels usuaris en forma de gràfic al llarg dels últims dies. Un detall important és que proveeix i configura automàticament el certificat SSL de la web, necessari si en algun moment l'aplicació es volgués convertir en una PWA, ja que és requisit fonamental per al seu funcionament.

5.2.2 Firebase Authentication

Aquest servei de Firebase dóna suport a tot el procés d'autenticació d'usuaris, el seu registre i accés a serveis de l'aplicació. L'objectiu és facilitar la creació de sistemes segurs que permetin verificar la identitat de forma fiable d'extrem a extrem i compatible

amb qualsevol plataforma. Integra diferents mètodes d'autenticació preestablerts amb diversos proveïdors d'identitat, principalment *Google*, *Facebook*, *Twitter*, *GitHub*, verificació telefònica o correu electrònic i contrasenya.

Mitjançant la consola, podem veure la llista completa d'usuaris de la web, amb les dates de registre i últim inici de sessió i el seu identificador alfanumèric únic, , desactivar i/o eliminar comptes i enviar correus de restabliment de contrasenya a usuaris en concret, activar i desactivar certs proveïdors d'identitat o modificar els correus electrònics automatitzats en la gestió d'usuaris.

A més ofereix una API completa per gestionar aquestes utilitats en la nostra aplicació, que en aquest projecte s'utilitza per implementar un servei de registre i inici de sessió basat en correu electrònic i contrasenya bàsic, que també permet verificar el correu i canviar les dades d'inici de sessió.

5.2.3 Firestore Database

Es tracta d'un servei de base de dades *Not only SQL*, una tecnologia cada cop més utilitzada basada en un model no relacional, que permet treballar amb estructures de dades més flexibles i una millor escalabilitat. Firestore treballa amb una estructura jeràrquica de documents o agrupacions de documents, anomenades col·leccions, i permet emmagatzemar, sincronitzar i consultar dades compartides en qualsevol dispositiu que hi tingui accés. Podem restringir aquest accés mitjançant una sèrie de normes completament personalitzables, que normalment es relacionen amb el sistema de gestió d'usuaris i els seus rols dins de l'aplicació, per exemple.

En la consola de Firebase Database es dóna accés total a la base de dades, tant per visualitzar-la com per editar el contingut d'aquesta, afegir atributs, editar camps, esborrar documents o col·leccions, etc. Permet també indexar els documents i col·leccions de forma personalitzada, per facilitar les cerques en un futur, útil sobretot si la quantitat de dades a emmagatzemar és molt alta o creix a mesura que escala l'abast de l'aplicació. També ens mostra una gràfica del tràfic de consultes i lectures a la base de dades al llarg del temps i permet editar les regles de seguretat utilitzades de controlar l'accés a aquesta informació. Actualment en la fase de desenvolupament l'accés a aquesta informació està obert per qualsevol usuari, però, com es comentara més endavant, és recomanable limitar l'accés en el moment de desplegar l'aplicació.

Des del codi de l'aplicatiu podem realitzar les operacions bàsiques d'una base de dades estàndard, com escriure, llegir, actualitzar o eliminar conjunts de dades. En la web desenvolupada, això s'utilitza per emmagatzemar tota la informació en un total de cinc col·leccions:

- Informació addicional d'usuaris - Firebase Authentication ofereix la possibilitat de gestionar els accessos i altres detalls d'interès, però per tal de mantenir un "perfil" de cada usuari amb informació d'utilitat el guardem en una col·lecció personalitzada.
- Productes - Informació de cada producte, amb el seu nom, una descripció, el preu, categoria a la qual pertanyen.
- Comandes - Informació de cada comanda en línia registrada amb la web, amb informació com l'identificador de l'usuari que l'ha creat, la llista de productes i quantitats que conté, l'hora de recollida i el preu total.
- Reserves - Conté els detalls de cada reserva realitzada, identificador de l'usuari, nombre de persones, dia, hora i taula concretament.
- Taules - Conté la llista de les taules disponibles, amb el seu nom, capacitat i torns disponibles. S'ha decidit implementar-ho d'aquesta manera per donar la possibilitat d'afegir més taules en un futur i que el funcionament del mòdul de reserves s'adapti automàticament.

5.3 Angular

En aquest apartat, s'explicarà el desenvolupament del codi font del projecte en el framework Angular, donat que es tracten d'uns quants centenars de línies de codi, s'explicarà el funcionament general de les diferents parts, adjuntant els trossos de codi quan es cregui convenient.

En primer lloc, hem de conèixer una mica el framework del que estem parlant. Angular és un framework per al desenvolupament d'aplicacions web, mantingut per Google però de codi lliure i basat en TypeScript, un codi "superset" de JavaScript, que bàsicament afegeix el tipatge de dades estàtic, però manté l'essència de JavaScript, tant en la sintaxi com en el procés de compilació.

Els elements fonamentals de codi en Angular s'anomenen components i són parts de codi reutilitzables que conformen l'aplicació. Un component està format per quatre arxius:

- component.ts - Arxiu *TypeScript* que descriu la seva funcionalitat.
- component.spec.ts - Arxiu que conté els tests unitaris del component, per facilitar la fase de testeig general de l'aplicació.
- component.html - Arxiu que descriu la seva interfície o estructura en llenguatge *html*.
- component.css - Arxiu d'estil en llenguatge *CSS* que defineix l'aspecte visual del component i dels seus tags *html*.

Un component pot ser des d'una vista de pàgina sencera de la web fins a la representació d'un element simple com podria ser un producte. Els components poden contenir altres components i anar construint així la web i la seva vista.

Angular inclou també una categoria especial, semblant a les classes, anomenada serveis que té com a objectiu gestionar correctament totes les accions o tasques en algun propòsit específic i amb la que els components interactuen, delegant aquelles tasques en les quals els serveis se centren. Utilitza la injecció de dependències [7] pròpia d'Angular per estar disponible per a qualsevol component.

Angular també treballa a nivell de mòduls, que contenen trossos de codi que agrupen components, directives, serveis, etc. que estan relacionats entre si. Els mòduls poden ser reutilitzats en altres projectes o combinar-se entre ells per atorgar funcionalitats, com si es tractessin de llibreries o classes.

L'aplicació desenvolupada consta d'un únic mòdul personalitzat que actua com a mòdul arrel del projecte, però utilitza mòduls predefinits en el propi framework per utilitzar certes funcionalitats. També s'han desenvolupat tres serveis diferents, el d'Autenticació, destinat a controlar les tasques relacionades amb els usuaris i l'API de Firebase Authentication, el servei dedicat a les comandes, que s'encarrega de gestionar les tasques del sistema de comandes, i per últim el servei de Dades, destinat a realitzar totes les accions relacionades amb el control de dades o variables i l'API de Firestore Database. També s'han desenvolupat un total de 14 components personalitzats diferents, dels quals es detallaran aquells dedicats a les funcionalitats o vistes que es consideren més importants.

5.3.1 Servei de dades

Aquest servei és probablement dels tres el més important, ja que és fonamental per tal de fer servir la majoria de funcionalitats de la web.

S'encarrega d'interactuar amb la Firestore Database, on es troben les diferents dades necessàries a recuperar quan naveguem per la web, com podrien ser els productes disponibles. De la mateixa manera, quan un usuari es registra o realitza una nova comanda/reserva, aquest servei és l'encarregat d'actualitzar la base de dades per tal que hi quedi constància i es pugui consultar més tard per part del mateix usuari o el client.

Les operacions de lectura i escriptura en aquesta base de dades es realitzen de forma asíncrona i s'ha de tenir en compte a l'hora de programar, ja que sinó pot comportar errors i no obtenir el funcionament esperat. Per tal d'evitar en part aquests errors, *Firestore* dona la possibilitat de treballar amb *Observables*, objectes que notifiquen a una llista d'altres objectes coneguts com a "subscriptors" quan pateixen algun canvi. Això permet per exemple, actualitzar automàticament la llista de comandes que es veuen en l'aplicació en el moment que se'n crea una de nova, de forma asíncrona i sense cap tipus d'error.

En el fragment de codi de la figura 1 podem veure en la línia 37 un exemple d'una funció d'aquest servei que configura un objecte subscriptor per a l'observable de la llista de productes de la base de dades, en la línia 49 veiem el seu ús per inicialitzar l'objecte *productList*, que es comentarà més endavant per mostrar la seva utilització. A més a més en les línies 45 i 46 podem veure la declaració de la injecció de dependències pròpia dels serveis d'angular que es comenta en l'anterior apartat.

```
37 | const getProductList = (collection: AngularFirestoreCollection<Product>) => {
38 |   const subject = new BehaviorSubject<Product[]>([]);
39 |   collection.valueChanges({ idField: 'id' }).subscribe((val: Product[]) => {
40 |     subject.next(val);
41 |   });
42 |   return subject;
43 | }
44 |
45 | @Injectable({
46 |   providedIn: 'root'
47 | })
48 | export class DataService {
49 |   productList = getProductList(this.store.collection('products'))
50 | }
```

Fig. 1: Exemple d'utilització d'Observables i injecció de dependències en el servei de dades

5.3.2 Mòdul/component arrel

També anomenat *AppModule* per convenció en el framework angular, és el mòdul principal de l'aplicació, d'on penja l'estructura formada de Mòduls preestablerts d'Angular, components i serveis personalitzats que conformen l'aplicació.

A part de la declaració d'aquesta estructura, conté el component bàsic sobre el que es muntarà la pàgina, anomenat *AppComponent*, aquest ja conté l'arxiu HTML amb el que és el codi bàsic de la pàgina web, l'estructura del títol, la barra de navegació, la secció de contingut i el peu de pàgina. La secció de contingut s'implementa a partir d'un mòdul preestablert d'Angular anomenat *AppRoutingModule* que s'encarrega de gestionar la navegació entre vistes de la pàgina web, renderitzant el component associat a cada *path* sota el tag HTML *<router-outlet>*.

Aquest element utilitza una *media query* en l'arxiu CSS, que mostra un estil de barra de navegació més compacte, que col·lapsa les diferents seccions en un únic botó desplegable, quan la pantalla del dispositiu de visualització és inferior als 450 píxels, per tal d'adaptar-ne l'ús en un dispositiu mòbil.

5.3.3 Component de Registre/Login/Perfil

Aquest component implementa tres funcionalitats diferents:

- Component de Registre - Ofereix la funcionalitat de registrar un nou usuari en la web. Consisteix bàsicament en un formulari on s'introdueix el correu electrònic, la contrasenya i el nom d'usuari a registrar. Després el component agafa les dades associades als camps i dona d'alta el nou usuari a la plataforma Firebase Authentication utilitzant el servei personalitzat d'autenticació i en Firebase Database amb el servei de dades detallat en l'anterior apartat. En cas que les credencials ja estiguin utilitzades, no siguin del format vàlid o altres errors, es mostra un missatge amb els detalls de l'error. En un principi, era accessible per a qualsevol visitant de la pàgina, per raons que es comenten en els pròxims apartats

finalment es reserva el seu ús als comptes d'administrador i per tant només aquests el podran veure i accedir-hi.

- Component d'Inici de Sessió - Ofereix la funcionalitat d'identificar-se amb les credencials d'un usuari registrat. Consisteix en un formulari que utilitza el servei d'Autenticació i l'API de Firebase Authentication per validar l'usuari i contrasenya. Si l'accés es concedeix, s'activa la vista d'usuari registrat, en la que podrà accedir a més funcionalitats i informació de la web. En cas de no concedir-se l'accés, es mostren els motius en un missatge d'error.
- Component de Perfil - Quan un usuari està identificat, es mostra la informació del seu perfil, com el seu nom d'usuari. Aquí es poden mostrar més detalls del perfil en un futur, com podria ser una imatge de perfil, una breu descripció de l'usuari o els punts i recompenses acumulats. També permet a l'usuari sortir de la seva sessió, així com accedir a un formulari per canviar les dades d'inici de sessió. Aquest formulari actua de forma semblant al del component de registre, però en lloc de donar d'alta un nou usuari, modifica les credencials de l'usuari identificat.

5.3.4 Component pàgina de Reserves

Aquest component incorpora les funcionalitats dedicades als usuaris registrats que vulguin realitzar la reserva d'una taula. En primer lloc la vista mostra la llista de reserves de l'usuari registrat. Ho fa combinant una operació de lectura de la col·lecció de reserves de la base de dades juntament amb una cerca i filtratge d'aquesta a partir de l'identificador d'usuari únic. La nova col·lecció temporal permet carregar en temps d'execució el component encarregat de mostrar els detalls d'una reserva en concret, tants cops com reserves ha fet l'usuari. En la figura 11 podem veure com es recupera un observable que conté la llista filtrada segons el camp d'usuari. En la figura 10 es mostra com renderitzar una llista de components amb la directiva **ngFor* a partir d'un observable, en aquest cas la variable `userBookList` guarda el valor retornat per la funció de la figura 11.

També mostra un botó que permet començar una nova reserva, i es demanen els detalls omplint un formulari reactiu, que varia els possibles valors dels seus camps en funció de la taula que s'ha escollit, ja que les disponibilitats i capacitats són diferents. Per escollir la data de la reserva, utilitza un mòdul predefinit que implementa el component *Date-picker*, aquest incorpora un calendari desplegable que podem personalitzar filtrant les dates vàlides amb els paràmetres adequats. Per exemple de moment només és possible seleccionar una data que sigui dissabte i amb un màxim d'un mes d'antelació, això pot canviar-se fàcilment en cas de ser necessari en un futur.

5.3.5 Component pàgina de productes

En aquest component, es mostra la llista dels productes disponibles. Utilitza un patró molt semblant al que implementa el component de reserves, adaptat al tipus de dades que recupera (productes en aquest cas en lloc de reserves), per tal de carregar el total de productes en temps d'execució, amb el seu nom i descripció.

A més a més, comprova si l'usuari actual està identificat a partir del servei de dades i en cas afirmatiu canvia la funcionalitat de cada component que representa un producte. Gràcies a això, un usuari sense identificar només veurà la llista normal dels productes, en canvi un usuari identificat, podrà gestionar una comanda en forma de cistell i un usuari administrador podrà afegir i editar els productes disponibles.

Per tal de gestionar la comanda en forma de cistell, es crea una instància d'un objecte comanda en el moment que l'usuari entra a la pàgina, gràcies a un *lifecycle hook*¹.

¹Els *lifecycle hook* són funcions especials que Angular ofereix per tal

5.3.6 Component detalls del producte

Aquest component és l'encarregat de renderitzar una representació visual de cada producte, sota una plantilla HTML personalitzada. Aquesta plantilla varia en funció de si l'usuari està identificat, fa aquesta comprovació a partir del servei de dades com altres components detallats anteriorment. Si no hi ha identificació es mostra únicament el nom i una descripció del producte. En canvi si hi ha identificació, es mostren els mateixos detalls, i s'afegeix el preu del producte i la funcionalitat d'emetre un *event*². A partir d'aquest *event*, que s'activa quan l'usuari fa doble clic en un producte, apareix un *pop-up* on es pot afegir al cistell, indicant primer la quantitat desitjada, cancel·lar l'operació o passar a finalitzar la comanda, moment en el qual es demanarà especificar l'hora de recollida i confirmació. Un cop confirmada la comanda, aquesta apareixerà en la pàgina de comandes de l'usuari.

5.3.7 Component pàgina de comandes i diàleg de nova comanda

En aquesta vista es mostra la llista de les comandes realitzades per l'usuari, aplicant una cerca i un filtre de la mateixa manera que en la pàgina de reserves, per tal de mostrar cada comanda, utilitza un component que renderitza els seus detalls, d'igual manera que ho fa el component de detalls del producte, però sense cap funcionalitat addicional en aquest cas.

També es pot iniciar la creació d'una comanda de forma diferent a la comentada anteriorment. En aquesta segona opció, es mostra un diàleg en forma de *pop-up* que conté un formulari on podem detallar tota la comanda de forma més centralitzada, sense necessitat d'anar movent-nos per la carta de productes o entrant i sortint dels *pop-ups* del cistell. S'utilitza *Property Binding*³ per mostrar un camp de selecció de producte, on el desplegable conté totes les opcions de productes de forma molt més reduïda. Un cop seleccionat el producte a afegir d'entre la llista, s'indica la quantitat i es pot afegir a la comanda.

Aquest procés es pot repetir tantes vegades com sigui necessari per afegir tots els productes sense tancar aquest desplegable i finalment confirmar la comanda, que es veurà reflectida en la mateixa pàgina.

5.3.8 Pàgina d'administrador

Aquesta vista només és accessible per aquells usuaris que tenen el rol d'administrador.

Actualment només es pot modificar aquest rol als usuaris des de la mateixa base de dades, modificant el document de l'usuari a concedir aquests permisos. Des d'aquesta pàgina, el client podrà accedir a les llistes completes de comandes i reserves, sense importar l'usuari que les hagi creat, per tal d'actuar en conseqüència com seria preparant la comanda en qüestió o validant la reserva. Permet filtrar la vista per mostrar aquells components que interessin en el moment.

5.3.9 Components de diàleg

Aquest tipus de components s'utilitzen per implementar algunes funcionalitats de la web, s'utilitza el mòdul d'Angular predefinit *MatDialogModule* per mostrar un diàleg en forma de *pop-up* per sobre de la vista actual de la web. En aquest diàleg es carrega la

de controlar les dades i funcions de les instàncies d'un component en moments clau al llarg del seu cicle de vida

²Un *event* es configura per escoltar i respondre a la interacció de l'usuari, com per exemple un clic o premer una tecla

³Property Binding permet donar valor a les propietats dels elements HTML en temps d'execució

plantilla HTML personalitzada del Component a mostrar i s'injecten les dades en forma d'objecte per tal de gestionar les funcions que realitzen de forma correcta.

El component `add-to-order`, encarregat d'afegir productes al cistell, és un clar exemple d'aquest tipus de components, en la figura 2 podem veure com instanciem aquest component dins d'un *pop-up* i declarem les dades que hi volem injectar. En la figura 3 es mostra la declaració del component i podem veure la injecció de les dades en la línia 24, en la figura 26 veiem la seva plantilla HTML (reduïda) i en la figura 25 podem veure la declaració de les seves funcions TypeScript, que controlen les funcionalitats dels botons.

```
addToOrder(product: Product){
  const dialogRef = this.dialog.open(AddToOrderComponent, {
    width: '500px',
    data: {
      product,
    }
  });
}
```

Fig. 2: Instanciació del diàleg d'afegir al cistell

```
17 export class AddToOrderComponent implements OnInit {
18   order: Order;
19   confirm: Boolean;
20   private backupOrder: Partial<Order> = { ... this.orderService.getCurrentOrder() }
21   constructor(public dialogRef: MatDialogRef<AddToOrderComponent>,
22     private dataService: DataService,
23     private orderService: OrderService,
24     @Inject(MAT_DIALOG_DATA) public data: AddToOrderData
25   ) {}
26   ngOnInit(): void {
27     this.confirm = false;
28     console.log("CURRENT ORDER: " + this.orderService.getCurrentOrder())
29     this.order = this.orderService.getCurrentOrder()
30   }
}
```

Fig. 3: Declaració del component add-to-order

5.4 Cascading Style Sheets (CSS)

En aquest apartat es comenta el desenvolupament del projecte en l'aspecte visual, en concret el codi CSS utilitzat per crear la interfície gràfica que es mostrarà en l'apartat de resultats. Detallar tot el codi CSS seria un document massa extens, per això es comentaran les pràctiques generals aplicades:

- **Ús de la propietat `display: grid`** - S'ha utilitzat l'eina *Grid* de CSS per tal d'estructurar gran part de l'aplicació. Aquesta eina divideix l'espai de l'element a estilitzar en una quadrícula, que podem definir i manipular per tal d'adaptar-la a les necessitats de l'aplicació. Permet treballar amb unitats adaptables a les dimensions de la pantalla de visualització, que permet crear aplicacions per qualsevol tipus de dispositiu de forma senzilla. Un aspecte important és que suporta la creació d'elements *grid* dins d'altres elements *grid*, això facilita sobretot els aspectes de posicionament dels elements en la vista de la pàgina un cop es comprenen els principis de funcionament. La conferència de *Bill Odom* sobre aquest tema [8] ha estat de gran ajuda en aquest sentit.
- **Estructura de la pàgina** - L'estructura de la vista de la web s'ha basat en un *Holy Grail Layout*⁴, implementat en un *grid*, amb alguns canvis. Per exemple s'ha prescindit de la secció de peu de pàgina perquè s'ha considerat innecessària fins al moment, tot i que l'estructura contempla afegir-la en un futur. També s'ha decidit implementar el menú o barra de navegació en la part de capçalera de la web o *header* i que aquesta quedi fixa a la part superior de la pàgina en tot moment.

⁴És un patró de disseny de la vista d'una pàgina web, molt comú en la xarxa però difícil d'implementar fins a l'aparició d'eines com *CSS grid* o *CSS Flexbox*

- **Ús del selector `:host`** - Angular proporciona aquesta eina per seleccionar el tag HTML propi del component, això facilita afegir estils a l'hora de definir la interfície gràfica de la vista.
- **Definició de classes** - Es defineixen un conjunt de classes en cada tag HTML a estilitzar, per tal de reutilitzar els estils en tots els elements d'una sola classe. Això permet per exemple estilitzar tots els botons d'una mateixa classe en tres línies de codi.
- **Gestió del desplaçament vertical** - Aquest aspecte és important en aquelles pàgines de la web en què el contingut a mostrar supera l'altura de la pantalla. En aquests casos, s'ha fet ús de la propietat *overflow: hidden* en un element pare, que permet indicar que s'amagui el contingut que no hi cap en l'element, juntament amb un element fill que conté la propietat *overflow: scroll*, que permet el desplaçament vertical al llarg del contingut. Això afegeix dues barres de desplaçament vertical, que molesten a l'hora de navegar per la web, però les amaguem amb les diferents propietats que es mostren en la figura 4, on cada una és suportada per un navegador en concret i s'afegeixen totes per oferir la mateixa vista en tots. D'aquesta manera obtenim un aspecte més agradable al mateix temps que mantenim la funcionalitat desitjada.

```
.main::-webkit-scrollbar /*Chrome, Safari i Opera*/
  display:none;

.main::-moz-scrollbar {
  width: 0px;
  height: 0px;
}

.main{
  scrollbar-width: none; /*Firefox*/
  -ms-overflow-style: none; /* Internet Explorer i Edge */
}
```

Fig. 4: Propietats per amagar la barra de desplaçament, segons navegador

5.5 Evolució del projecte

5.5.1 Evolució de requisits

- **Diferenciar usuaris registrats de no registrats** - Es demana fer així degut a la naturalesa de l'experiència oferta pel client, que de moment se centra en un públic molt concret en les primeres fases, com són persones conegudes, amics i familiars del client.
- **Exclusivitat en el registre** - Per tal de portar un control sobre l'abast de l'experiència i el seu públic, només s'ha de permetre el registre a aquelles persones que el client consideri, per mantenir l'ordre, evitar usuaris malintencionats i mantenir una certa "exclusivitat", a més de controlar els aforaments i contactes a causa de les mesures sanitàries del moment.

5.5.2 Canvis en la planificació

Al llarg del desenvolupament, hi ha hagut diversos factors que han fet sorgir la necessitat de variar la planificació inicial del projecte, a continuació es destaquen els més importants juntament amb les seves causes, conseqüències i altres observacions.

- **Reestructuració del mòdul de comandaments** Un cop finalitzada la fase detallada en el diagrama com a **tasca 27. Desenvolupament - Comandes Online**, en la que s'havia desenvolupat el sistema de comandaments en línia, les expectatives del client no concordaven amb el resultat obtingut, el mètode a seguir per tal de realitzar una comanda requeria massa "esforç" per part

de l'usuari, ja que s'havia de realitzar en una pàgina diferent a la dels productes oferts.

En el nou sistema de comandes proposat, se segueix un model de comanda amb cistell on l'usuari pot anar afegint-hi ítems des de les mateixes descripcions dels productes i es permet tancar la comanda en qualsevol moment, passant a ultimar els detalls necessaris i finalment confirmar la comanda. Aquesta reestructuració del sistema suposa el primer endarreriment del desenvolupament real del projecte respecte a la planificació inicial i comporta una modificació de prioritats en les últimes fases, on les tasques **tasca 23. Pla de fidelització dels usuaris** i **tasca 24. Part Social de l'aplicatiu** perden en part la seva prioritat en augment de la d'aquesta.

- **Retràs sistema de reserves** El mòdul de gestió de reserves és la segona causa principal d'endarreriment en el transcurs del projecte. Degut a un mal disseny inicial, a meitat de desenvolupament s'ha hagut de descartar el treball realitzat durant els primers dies d'aquesta fase, i s'ha procedit a repensar l'estructura del codi i el funcionament d'aquest, per tal que complissin amb els requisits del client. En un principi s'havia pensat dissenyar un sistema de reserves basat en la comunicació entre usuaris i client, per tal d'acordar els detalls de la reserva, ja que fins al moment era el procediment habitual en aquests casos. Finalment, el client demanava un sistema que no depenguis tant de la interacció amb l'usuari, ja que això complicava i ralentitzava el procés. Per tant el sistema de reserves que s'ha desenvolupat pren una sèrie de paràmetres proporcionats pel client per mostrar la disponibilitat en el moment i agilitzar el sistema.
- **Incorporació fase de desenvolupament visual** En la planificació inicial, no hi havia una fase dedicada al disseny visual de l'aplicació. En un primer moment es considerava que era una tasca que es realitzaria de forma conjunta en cada fase del desenvolupament funcional. Més tard, s'ha arribat a la conclusió que aquest aspecte era més complex de desenvolupar del que es tenia en compte i que per tal de fer-ho de forma més correcta, clara, organitzada i concisa, s'hauria de dedicar més temps del esperat i de forma aïllada del desenvolupament funcional. Es creu que separant aquestes dues parts, es podrà enfocar millor el treball en cada moment concentrant els processos d'aprenentatge i búsqueda de conceptes o fonts d'informació en cada tema.
- **Eliminació fase de testeig** En un principi, es tenia pensat desplegar un prototipus de l'aplicació en un entorn real, una mica controlat, on es poguessin identificar els punts forts i febles de l'aplicatiu en les seves fases finals i per poder rebre, encara que fos en petita mesura, un *feedback* d'usuaris reals en el moment d'utilitzar l'eina per els seus usos funcionals principals, com podrien ser realitzar reserves, comandes, veure informació dels productes, etc. Degut a com ha anat evolucionant la situació, s'ha vist poc a poc que això era inviable, principalment perquè la situació del moment ha dificultat molt el fet de obrir aquesta experiència oferta pel client a un públic suficientment gran per a obtenir dades útils, però també perquè finalment els temps de desenvolupament s'han allargat més de l'esperat.
- **Endarreriment de la fase de desplegament** Degut també a l'endarreriment dels temps de desenvolupament, la part final del desenvolupament del projecte tampoc s'ha pogut realitzar tal i com s'havia planificat en un principi, però en un altra secció més endavant es detallarà juntament amb altres línies de continuació a seguir en el desenvolupament del projecte que queden fora de l'abast d'aquest treball a causa del marge de temps.

Aquelles fases que no es comenten en aquest apartat han seguit la

seva planificació inicial de forma adequada i sense canvis destacables.

6 RESULTATS

Aquest apartat té com a objectiu exposar i interpretar els resultats obtinguts al llarg del desenvolupament del projecte.

Com a principal resultat destaca el que és l'aplicatiu en general, que s'ha aconseguit desenvolupar en gran aspecte, tant en la part funcional com en la part visual, a partir de coneixements previs com a estudiant d'enginyeria informàtica i amb els coneixements obtinguts durant el desenvolupament del projecte, tant en assignatures paral·leles com a partir de la recerca personal per la web.

6.1 Aplicació

L'aplicació resultant és la que obtenim a partir de la combinació del codi font en Angular, les fulles d'estil en CSS i el *back-end* en Firebase. La barra de navegació es troba fixe en la capçalera de la pàgina en tot moment, però va canviant el seu aspecte en funció de la situació.

En la figura 12 es pot veure la barra de navegació per a un usuari identificat, en canvi en la figura 13 es mostra la mateixa barra de navegació visible per a un usuari un cop s'identifica. En la figura 5 s'observa la vista en un dispositiu mòbil.



Fig. 5: Barra de Navegació d'un usuari de dispositiu mòbil

A partir d'aquí tenim les diferents vistes o pàgines de la web. En primer lloc la pàgina principal, on es mostra tota la informació rellevant, redactada pel mateix client, sobre l'experiència i a la que l'usuari pot tornar en qualsevol moment des de la pestanya d'Informació o la imatge de la barra de navegació. En la figura 14 podem veure part d'aquesta vista, organitzada en etiquetes amb un tros de text i una fotografia relacionada. En segon lloc, relacionada també amb l'aspecte informatiu de la web, la pàgina de productes, que ofereix una descripció dels productes disponibles en funció de l'estat de la sessió de l'usuari. La vista de la figura 6 es mostra per als usuaris identificats, mentre que als que no ho estan es mostra la mateixa vista, però sense l'etiqueta que conté el preu, a més de la variació de les funcionalitats, que ja s'han comentat en l'apartat de desenvolupament.

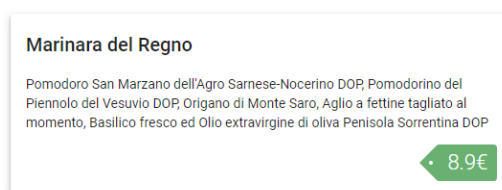


Fig. 6: Vista de detalls del producte d'un usuari identificat

Aquestes dues pàgines compleixen l'objectiu d'oferir una pàgina de presentació i informació de l'experiència, i es considera

que el resultat visual final és adequat per al projecte.

El resultat de la pàgina de contacte varia una mica en relació amb l'objectiu plantejat, ja que es volia oferir una pàgina completa dedicada a aquesta funcionalitat, però finalment s'ha implementat un diàleg en forma de *pop-up* que ofereix els detalls de contacte d'interès per als clients, que són bàsicament el número de contacte, correu electrònic, Instagram i, en cas que l'usuari s'identifiqui, l'adreça. Es considera que aquest canvi no afecta en gran manera, perquè la finalitat i funcionalitat es manté en la versió final. En la figura 7 podem observar aquest diàleg.



Fig. 7: Diàleg de contacte

El resultat del sistema de comandes en línia també ha variat una mica respecte a l'objectiu principal, però en aquest cas ha millorat, ja que, a part que compleix la funcionalitat requerida, es pot realitzar de dues formes diferents, com ja s'ha explicat en l'apartat de desenvolupament d'Angular d'aquest document. A més a més s'ofereix una pàgina dedicada a veure la informació relacionada amb les comandes, que podem veure en la figura 15 juntament amb els detalls d'una comanda de l'usuari. Per tant es considera que s'ha assolit l'objectiu plantejat amb un resultat fins i tot millor de l'esperat.

El resultat del sistema de reserves ha seguit en la línia del plantejat en els objectius finals, tot i haver necessitat un replantejament en alguna fase del seu desenvolupament per tal d'aconseguir funcionalitat desitjada. Podem observar el resultat de la pàgina de reserves en la figura 16, en la que es mostren les diferents reserves de l'usuari. En la figura 17 es mostra el formulari que permet aquesta funcionalitat i per tant compleix l'objectiu proposat inicialment.

A més a més, també s'han assolit alguns altres resultats que no formaven part dels requisits o objectius inicials, però que han sorgit a partir de l'evolució en el desenvolupament del projecte i el feedback rebut pel client. Principalment podem destacar el fet de poder distingir entre els usuaris registrats i aquells que no ho estan a partir del mòdul d'inici de sessió, i el control de les funcionalitats addicionals que obtenen els usuaris un cop es registren. Això es du a terme en la pàgina d'inici de sessió, també anomenada de perfil, segons l'estat sessió de l'usuari. La funcionalitat d'aquesta pàgina ja s'ha presentat en detall en la secció de desenvolupament, però en aquest apartat es presenten els resultats de la interfície d'usuari. Al principi, la vista d'un usuari sense identificar és bàsicament un formulari d'inici de sessió, la figura 18 mostra els detalls de l'usuari ja identificat, amb les opcions de tancar la sessió, editar les dades d'inici de sessió mitjançant el formulari de la figura 19 i l'accés a les opcions de crear noves comandes o reserves, totes funcionals. La secció de contingut de l'usuari és simplement per delimitar l'espai on es mostrarien futurs aspectes o funcionalitats relacionats amb l'usuari. En últim lloc, la vista de la figura 20 mostra la mateixa pàgina quan l'usuari té el rol d'administrador i

pot accedir a les funcionalitats de la vista d'administrador 21 i del registre de nous usuaris 22.

També s'ha desenvolupat un sistema d'administració, des del qual es pot gestionar diferents aspectes de l'aplicació, destaca principalment que aquests usuaris són els únics que poden registrar nous usuaris, per tal de mantenir l'exclusivitat i el control de l'abast de l'experiència, tal com es demana per part del client al llarg de l'evolució del projecte. A més a més, també permet gestionar la informació dels productes oferts, afegir nous productes, editar els existents o eliminar-los segons el seu criteri. Això es fa perquè el client pugui també gestionar aquest tipus d'informació des de la pròpia aplicació i no sigui necessari accedir a la consola del projecte de Firebase, que es reserva per al rol del desenvolupador. En la figura 23 podem veure el formulari per afegir un nou producte i en la figura 8 el diàleg per tal d'editar un d'existent.

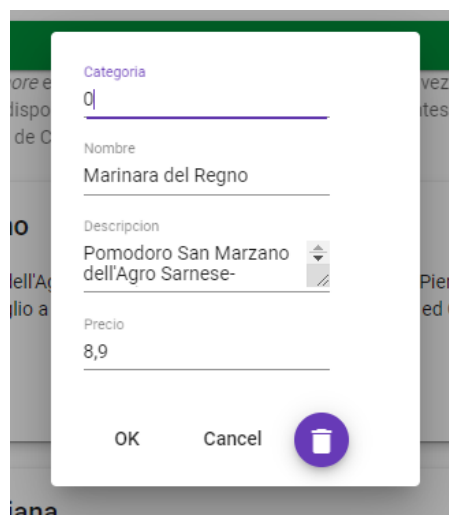


Fig. 8: Diàleg per editar un producte com a administrador

Es considera que amb la combinació dels dos últims resultats presentats, s'ha assolit la totalitat dels objectius i requisits principals, juntament amb bona part dels que han sorgit durant l'evolució del projecte.

7 LÍNIES DE CONTINUACIÓ

Aquest apartat pretén definir una sèrie de línies de continuació general, que s'han de tenir en compte per seguir amb el projecte a partir dels resultats que es presenten en l'anterior apartat.

7.0.1 Presents en la planificació

Per començar, aquells punts que s'havien definit o plantejat dins l'abast del projecte, però que per raons de temps o de l'evolució del projecte i el seu context no s'han pogut dur a terme.

- **Fase de Testeig** - L'endarreriment del desenvolupament en el projecte ha causat que quedi fora de l'abast d'aquest, es plantejava utilitzar un grup reduït d'usuaris de confiança com a *Beta testers* per provar el funcionament de l'eina, sobretot les funcionalitats reservades als usuaris registrats. Aquesta seria la següent fase per tirar endavant el projecte i es podria realitzar mitjançant l'eina *Test Lab* de la plataforma Firebase
- **Fase de desplegament** - Aquesta fase també s'ha descartat en l'evolució del projecte pel mateix motiu d'endarreriment de temps juntament amb la situació sanitària actual que dificulta l'accés a un grup d'usuaris real. Es podria dur a terme amb l'eina de *App Distribution* de Firebase, però primer s'hauria d'acabar de definir i portar a terme altres línies de mi-

llora com serien les normes de seguretat de la base de dades i la política de protecció de dades.

- **Progressive Web App** - Com a pas final del desplegament, seria interessant aplicar el patró d'una PWA a l'hora d'oferir l'aplicació, per acabar d'oferir-la de forma global en qualsevol tipus de dispositiu i oferir a l'usuari l'opció d'instal·lar-la com si es tractés d'una aplicació nativa. Hi ha un munt d'eines dedicades a aquest fi, però el més convenient seria desenvolupar-ho amb els mòduls predefinits d'Angular que ofereixen serveis dedicats a aquesta conversió.

7.0.2 Sorgits del desenvolupament

En aquest apartat es detallen les línies de continuació i millora que no es contemplaven en l'abast del projecte inicialment, però que han anat sorgint a mesura que avançava el desenvolupament del projecte.

- **Protecció de dades** - En un principi no s'havien considerat necessari cap política de protecció de dades, perquè no es recopila pràcticament cap en aquest sentit, únicament el correu electrònic. Com el registre està reservat només als usuaris de confiança, pel moment s'entén que donen la seva aprovació per gestionar-los. De fet actualment, no és necessari validar el correu electrònic per utilitzar l'aplicació i per tant es poden utilitzar formats de correu que ni tan sols existeixen per donar accés als usuaris. Una línia de continuació important, seria establir una política de protecció de dades vàlida que contemplés la legislació vigent [10]. Actualment s'haurien de contemplar els següents punts:
 - Recopilar només aquella informació necessària.
 - Detallar l'ús que es fa d'aquesta informació, en el moment de demanar-la i posteriorment en alguna secció de protecció de dades.
 - Oferir l'opció de donar-se de baixa o esborrar la informació associada a un usuari en qualsevol moment.
 - Informar a l'usuari de la localització d'emmagatzemament de les dades, tan geogràfica com dins d'empreses o organitzacions (per exemple en el cas d'aquesta aplicació l'emmagatzematge de les dades es fa sota els serveis de Google i en la regió d'Europa Occidental)
- **Acces base de dades** - També s'haurien d'establir unes regles d'accés a la base de dades que gestionin l'accés a les dades segons el rol o identificador de l'usuari. De moment aquest accés es fa en el mateix servidor de l'aplicació mitjançant el servei de dades, però és recomanable passar aquesta gestió a les normes d'accés de *Firestore Database* en el moment de desplegar l'aplicació.
- **Millors d'adaptabilitat** - Per tal d'oferir una millor experiència en tots els tipus de dispositiu, s'hauria d'acabar de polir l'aspecte visual en certs tipus de dispositius, sobretot aquells que es troben a mig camí entre el tamany de mòbil estàndard i el tamany d'ordinadors de sobretaula estàndards.
- **Visuals** - Tot i que els resultats obtinguts es consideren prou bons, per haver-se desenvolupat en *css* bàsic i sense coneixements previs, seria recomanable seguir millorant aquesta interfície gràfica abans de desplegar l'aplicació en l'entorn real, per exemple una bona opció seria la de contractar una persona especialitzada perquè realitzi aquest disseny.

8 CONCLUSIONS

Es considera que el desenvolupament d'aquest projecte ha estat encertat i que els resultats són bons, tenint en compte que en el moment d'inici del projecte el coneixement i experiència en el desen-

volupament web era pràcticament inexistent, sobretot en quant a l'ús de les tecnologies i eines utilitzades.

Tot i això, un cop acabat, fent una mica d'anàlisi dels resultats i el desenvolupament del projecte, es creu que es podria haver desenvolupat una aplicació d'aspecte més professional i modern si s'hagués desenvolupat la web sota un sistema de continguts com *Wordpress*, que ofereix una gran quantitat d'eines i plantilles per realitzar aquest tipus de pàgines de forma ràpida i senzilla. En canvi, l'interès i el grau de dificultat en el projecte haurien perdut valor, a part que el procés d'aprenentatge i el desenvolupament d'una aplicació d'aquest tipus de forma lliure ha sigut interessant i enriquidor.

També s'ha assolit un cert grau d'experiència en l'àmbit del desenvolupament web i en l'interacció amb un projecte en un entorn similar al real, en quant a les tasques de planificació del projecte, el seu desenvolupament, la relació amb una figura de client, etc. Que es plantejaven com a objectiu del projecte i necessaris de cara al món real professional.

REFERÈNCIES

- [1] Wikipedia. *Progressive web application*. Consultat el Juny de 2021 [En línia] Disponible a: https://en.wikipedia.org/wiki/Progressive_web_application
- [2] Angular by Google, *Angular Development Docs*. Consultat el Maig de 2021 [En línia] Disponible a: <https://angular.io/docs>
- [3] Firebase by Google *Firebase User Guide* | Google. Consultat el Maig de 2021 [En línia] Disponible a: <https://firebase.google.com/docs/web/setup>
- [4] Wordpress. *Wordpress*. Consultat el Maig de 2021 [En línia] Disponible a: <https://wordpress.com/es/>
- [5] w3tech. *w3tech Usage statistics and market share of Wordpress*. Consultat el Maig de 2021 [En línia] Disponible a: <https://w3techs.com/technologies/details/cm-wordpress>
- [6] M.Tena *¿Qué es la metodología 'agile'?*. Consultat el Febrer de 2021 [En línia] Disponible a: www.bbva.com/es/metodologia-agile-la-revolucion-las-formas-trabajo
- [7] Angular by Google, *Dependency injection in Angular*. Consultat el Juny de 2021 [En línia] Disponible a: <https://angular.io/guide/dependency-injection>
- [8] B. Odom, *"Angular and CSS Grid: Get ready to fall in love"*. Consultat el Juny de 2021 [En línia] Disponible a: https://www.youtube.com/watch?v=1h6n0JxXD_g&ab_channel=ng-conf
- [9] W3schools.com. *2021. CSS Tutorial*. Consultat el Juny de 2021 [En línia] Disponible a: <https://www.w3schools.com/css/>
- [10] España. *Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales*. Consultat el 25 de Juny de 2021 [En línia] Disponible a: <https://www.boe.es/buscar/act.php?id=BOE-A-2018-16673>

A APÈNDIX

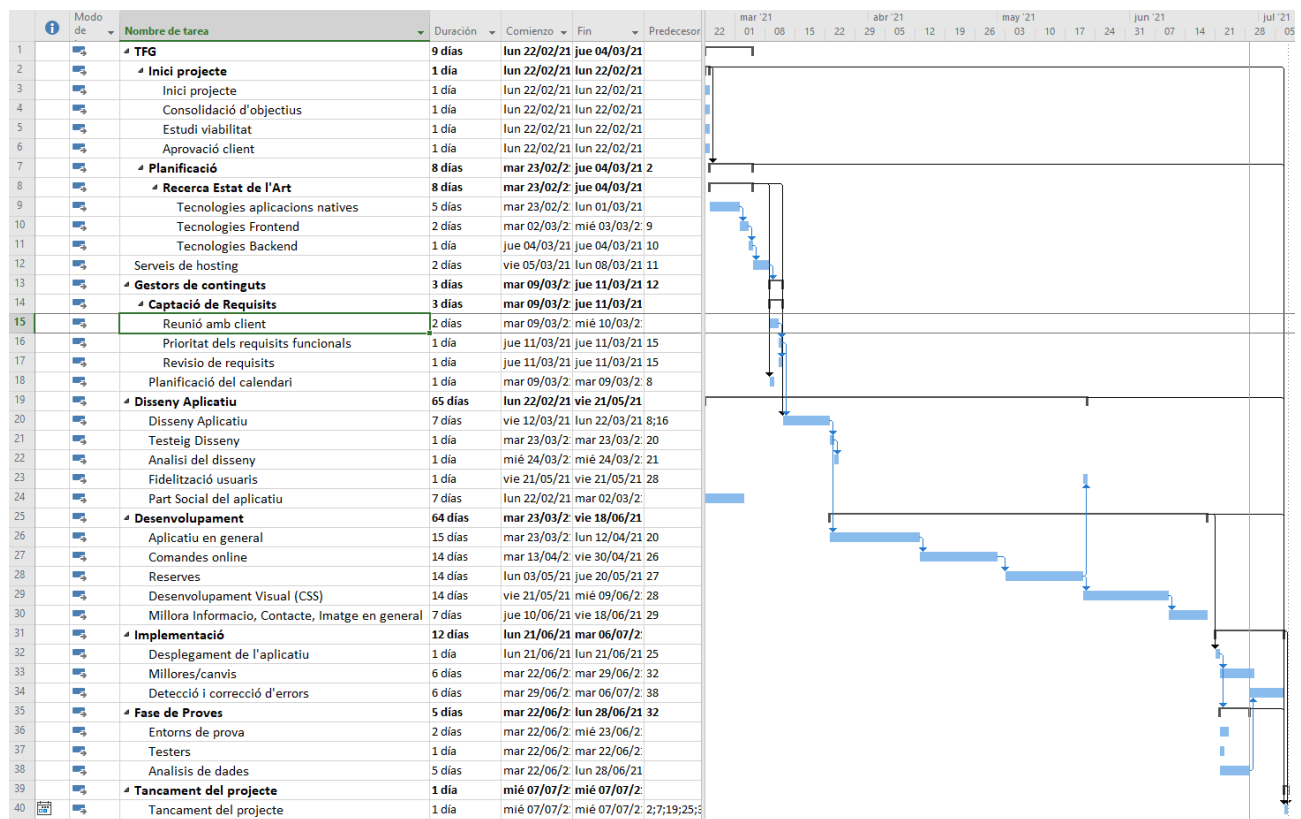


Fig. 9: Planificació inicial del projecte

```
<app-book-details *ngFor="let book of this.userBookList | async" [book]="book"></app-book-details>
```

Fig. 10: Exemple de ngFor per instanciar tants components com elements en una llista i dotar-los de valor

```
getUserBookList(userId: string) : BehaviorSubject<any[]>{
  let collection = new AngularFireStoreCollection<unknown>([this.store.collection('books').ref,
    this.store.collection('books').ref.where("uid", "==", userId), this.store])
  let filteredBooks = getUnknownList(collection)
  return filteredBooks
}
```

Fig. 11: Exemple de filtratge d'una llista de la base de dades, en concret segons el camp d'usuari



Fig. 12: Barra de Navegació de l'usuari sense identificar



Fig. 13: Barra de Navegació d'un usuari identificat



Fig. 14: Pàgina principal i d'informació de la web

PEDIDOS

Aquí puedes ver tu historial de pedidos con nosotros y crear nuevos pedidos

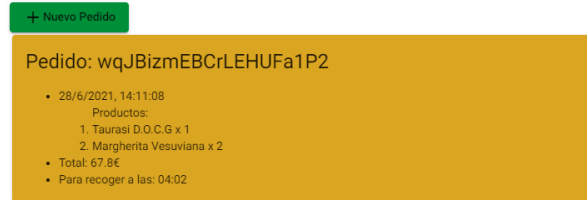


Fig. 15: Vista de pàgina de comandes amb una comanda

Reservas

Admitimos nuevas reservas con un mes de antelación

Aquí puedes consultar tus reservas con nosotros:

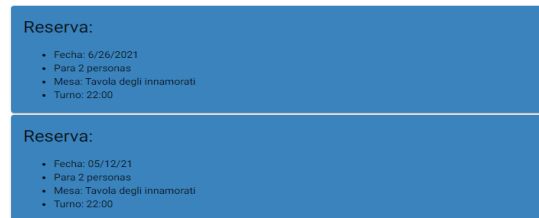


Fig. 16: Vista de pàgina de comandes amb dues reserves

Reservas

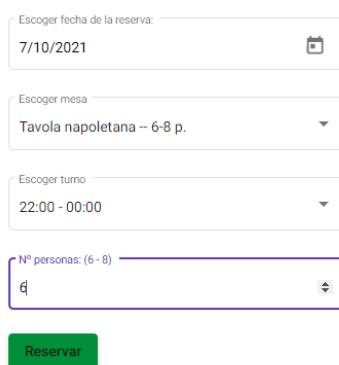


Fig. 17: Formulari de creació de reserves per part de l'usuari

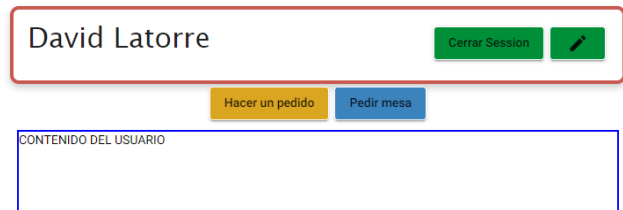


Fig. 18: Vista del perfil de l'usuari i les seves opcions

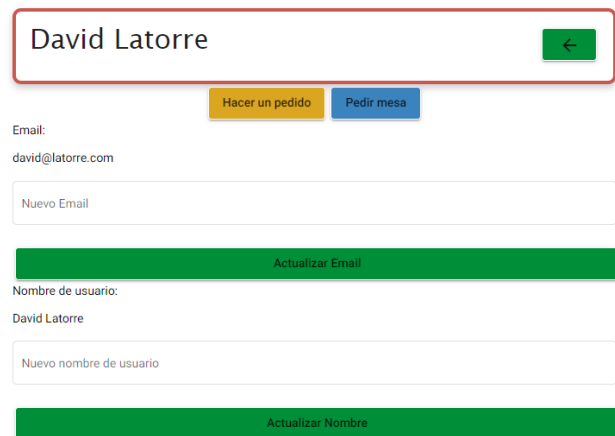


Fig. 19: Formulari de canvi de dades d'inici de sessió de l'usuari



Fig. 20: Vista del perfil d'un usuari administrador

Página d'administrador

Aquí se muestran las listas de Pedidos i/o Reservas de los usuarios de la web

Filtrar: ☒ Pedidos ☒ Reservas



Fig. 21: Vista d'administrador amb els dos filtres de detall activats

Registra

Email

Contraseña

Nombre de usuario

Registrar usuario

Fig. 22: Formulari de registre d'usuaris accessible com a administrador

Categoria

Nombre

Descripcion

Precio

OK Cancel

Fig. 23: Diàleg per crear/afegir un nou producte com a administrador

1		TFG	9 días
2		Inicio projecte	1 día
3		Inicio projecte	1 día
4		Consolidació d'objectius	1 día
5		Estudi viabilitat	1 día
6		Aprovació client	1 día
7		Planificació	8 días
8		Recerca Estat de l'Art	8 días
9		Tecnologies aplicacions natives	5 días
10		Tecnologies Frontend	2 días
11		Tecnologies Backend	1 día
12		Serveis de hosting	2 días
13		Gestors de continguts	3 días
14		Captació de Requisits	3 días
15		Reunió amb client	2 días
16		Prioritat dels requisits funcionals	1 día
17		Revisio de requisits	1 día
18		Planificació del calendari	1 día
19		Disseny Aplicatiu	65 días
20		Disseny Aplicatiu	7 días
21		Testeig Disseny	1 día
22		Anàlisi del disseny	1 día
23		Fidelització usuaris	1 día
24		Part Social del aplicatiu	7 días

Fig. 24: Tasques relacionades amb l'inici del projecte

```

addProduct(): void{
    this.order.productsList.push(this.data.product);
    this.order.amountList.push(this.data.amount);
    this.order.totalPrice += (this.data.product.price * this.data.amount)
    this.order.totalPrice = Math.round((this.order.totalPrice + Number.EPSILON) * 100) / 100
    console.log("NEW ORDER " + this.order)
}

endOrder(){
    const errorHtml = document.querySelector("div[id='addToOrder']");
    if(!this.confirm){
        this.confirm = true;
    }else{
        console.log(this.order.productsList.length)
        this.order.date = new Date().toLocaleString()
        if(this.order.productsList.length > 0){
            console.log(this.order.productsList.length)
            this.dataService.store.collection('orders').doc(sessionStorage.getItem('current_order'))
                .set({ productsList: this.order.productsList, amountList: this.order.amountList, totalPrice: this.order.totalPrice })
                .catch(err => {
                    console.log("ERROR: " + err)
                    errorHtml.querySelector('.error').innerHTML = "*ERROR* " + err.message ;
                })
        }
    }
}
}

```

Fig. 25: Funcions TypeScript del component add-to-order

```

<div id="addToOrder">
  <div class="big">...
  </div>
  <div class="inputField">
    <mat-form-field *ngIf="!this.confirm" class="amount">
      <mat-label>Cantidad</mat-label>
      <input type="number" matInput [(ngModel)]="data.amount">
    </mat-form-field>
    <div *ngIf="this.confirm" class="amount">
      <mat-form-field>
        <mat-label>Hora de Recogida:</mat-label>
        <input type="time" matInput [(ngModel)]="this.order.hour">
      </mat-form-field>
    </div>
  </div>
</div>

<div class="order">...
</div>

<div class="actions">
  <div mat-dialog-actions *ngIf="!this.confirm" class="actions2">
    <button mat-button class="primary addProduct" (click)="this.addProduct()">Añadir producto</button>
    <button mat-button class="primary back" [mat-dialog-close]="{ order: this.order }">Volver a la carta</button>
    <button mat-button class="secondary confirmOrder" (click)="this.endOrder()">Finalizar pedido</button>
  </div>
  <div mat-dialog-actions *ngIf="this.confirm" class="confirm">
    <button mat-button class="primary cancelConfirm" (click)="cancel()">Cancel</button>
    <button mat-button class="secondary endOrder" (click)="this.endOrder()" [mat-dialog-close]="{ order: this.order }">Finalizar pedido</button>
  </div>
</div>

<div class="right">...
</div>

<p class="error center-align" style="color: red"></p>
</div>

```

Fig. 26: Plantilla HTML del component add-to-order